

CLAIMS

What is claimed is:

1. A method for transferring CPU budget and CPU control between and client thread and a server thread in a client/server pair, comprising:

5 assigning a CPU budget to said client thread;
executing said client thread at a scheduled time within a first period;
transferring, within said first period, CPU control and any unused CPU budget to said server thread when said client thread stops executing;
executing said server thread within said first period; and
10 transferring, within said first period, CPU control and any unused CPU budget to said client thread when said server thread stops executing.

2. A method according to claim 1 further comprising alternately transferring CPU control and unused CPU budget between said client thread and said server thread within said
15 first period.

3. A method according to claim 2 further comprising terminating the execution of said client thread and said server thread when said CPU budget has expired.

20 4. A method according to claim 3 wherein the first step of executing comprises transferring service requests from the client to the server.

5. A method according to claim 4 wherein the second step of executing comprises transferring results of the service requests from the server to the client.

6. A method according to claim 5 wherein said client thread places service request in a client-to-server queue when said client thread is executing and wherein said server thread retrieves and processes the service request when said server thread is executing.

5 7. A method according to claim 6 wherein said server thread places the results of the service request in a server-to-client queue when the server thread is executing and wherein said client thread retrieves the results when said client thread is executing.

8. A method according to claim 7 wherein the first step of transferring occurs when
10 said client thread has completed sending service requests to said client-to-server queue.

9 A method according to claim 7 wherein the first step transferring occurs when said client-to-server queue is full.

10 15 10. A method according to claim 7 wherein the first step of transferring occurs when a service request must be processed immediately.

11. A method according to claim 7 wherein the second step of transferring occurs when said server-to-client queue is full.

20 12. A method according to claim 7 wherein the second step of transferring occurs when said server thread empties said client-to-server queue.

13. A method according to claim 7 wherein the second step of transferring occurs when
25 said server thread is responding to a priority service request from said client thread.

14. A method according to claim 7 wherein the first step of transferring occurs upon the occurrence of a synchronization object.

5 15. A method according to claim 14 wherein the second step of transferring occurs upon the occurrence of a synchronization object.

16. A method according to claim 15 wherein said synchronization object is an event.

10 17. A method according to claim 15 wherein said synchronization object is a semaphore.

18. A method according to claim 1 wherein the CPU budget assigned to said client thread is sufficient to complete the task of the client/server pair.

15 19. A method according to claim 1 further comprising assigning a CPU budget to said server thread.

20. A method for transferring CPU control between a client thread and a server thread in a client/server pair, comprising:

20 executing said client thread at a scheduled time within a first period;

transferring control of the CPU within said first period to said server thread when said client thread stops executing;

executing said server thread in said period; and

transferring within said first period, control of the CPU to said client thread when

25 said server thread stops executing.

21. A method according to claim 20 further comprising alternately transferring CPU control between said client thread and said server thread within said first period.

22. A method according to claim 20 wherein the first step of executing comprises
5 transferring service requests from the client to the server.

23. A method according to claim 22 wherein the second step of executing comprises transferring results of the service requests from the server to the client.

10 24. A method according to claim 23 wherein said client thread places service requests in a client-to-server queue when said client thread is executing and wherein said server thread retrieves and processes the service requests when said server thread is executing.

15 25. A method according to claim 24 wherein said server thread places the results of the service requests in a server-to-client queue when the server thread is executing and wherein said client thread retrieves the results when said client is executing.

26. A method according to claim 25 wherein the first step of transferring occurs when said client thread has completed transferring service requests to said client-to-server queue.

20 27. A method according to claim 25 wherein the first step of transferring occurs when said client-to-server queue is full.

28. A method according to claim 25 wherein the first step of transferring occurs when a
25 service request must be processed immediately.

29. A method according to claim 25 wherein the second step of transferring occurs when said service to client queue is full.

30. A method according to claim 25 wherein the second step of transferring occurs when said server thread empties said client-to-server queue.

31. A method according to claim 25 wherein the second step of transferring occurs when said server thread is responding to a priority service request from said client thread.

32. A method according to claim 25 wherein the first step of transferring occurs upon the use of a synchronization object.

33. A method according to claim 32 wherein the second step of transferring occurs upon the use of a synchronization object.

34. A method according to claim 33 wherein said synchronization object is an event.

35. A method according to claim 33 wherein said synchronization object is a semaphore.